Geometric Approach to Logistic Regression

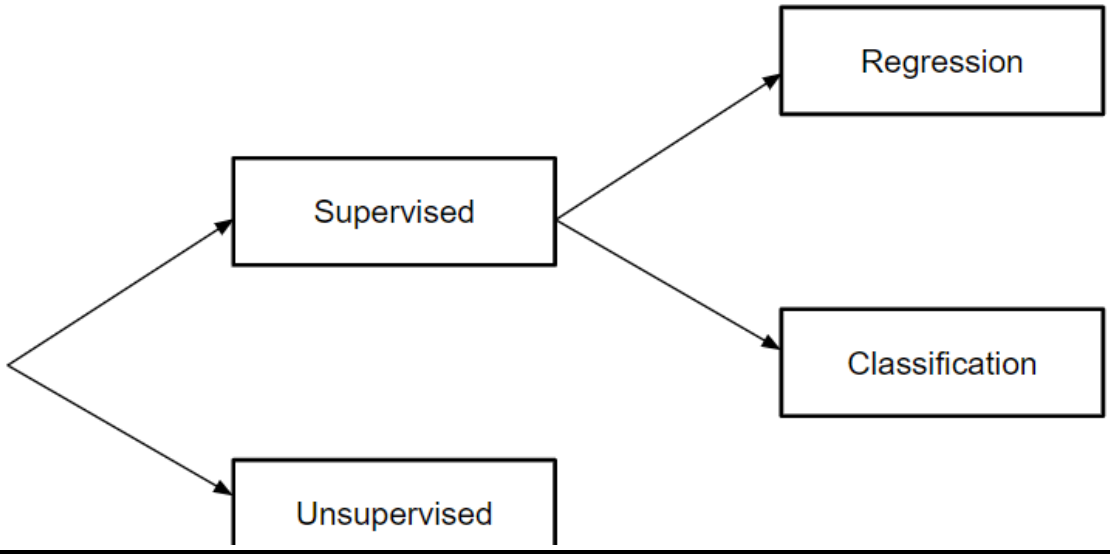
Praveen Hegde
201939, MSc Statistics 2nd SEM
SDM PG COLLEGE, UJIRE

CLASSIFICATION AND REGRESSION

Introduction:

In data science, we have two parts. Supervised learning and Unsupervised learning. Under supervised learning there are two main categories, one is **Regression** another is **Classification**.

Let us discuss about classification problems….



What is classification?

Imagine that you are a data analyst/scientist at Flipkart. Flipkart is an e-commerce company. You might have seen reviews about a product (say Samsung M31 mobile) in Flipkart. Some of the reviews are positive and some of the reviews are negative. If we read the review, we can say whether a review is positive or negative. But remember Flipkart is not a small company, it has millions of products and customers. As a data analyst, one cannot manually read all the reviews and classify them as positive or negative.

So there comes Machine learning.

If you teach computer to classify positive and negative reviews using reviews data that you have, then the classification task becomes cost effective, and also it takes less time compared to human. So, we should teach machine to learn the task of classification.

The main question is ***HOW TO TEACH THE MACHINE?***

Computer can learn equation, maths easily. So, we can model some rules to follow for classification based on the reviews data. These set of rules are models. Logistic regression is one of the classification models. It is mainly used for binary classifications. We can also extend this for multi class classification.

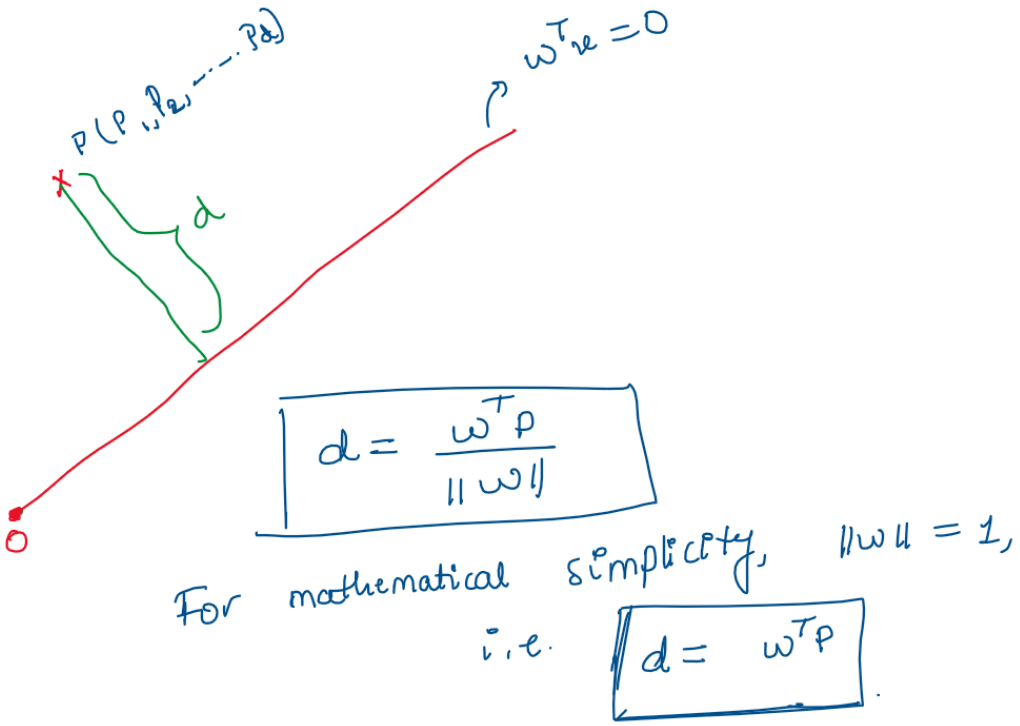
But let us understand Logistic regression for binary classification using geometric approach.

Let's consider same review example. So, we have 2 classes, positive and negative. Hence this is a binary classification task.

Given the reviews we cannot use that directly while modelling. We need to vectorize that. Vectorizing the data is not considered here because our main focus is to learn Logistic Regression.

Imagine that you have vector representation for each review, and we have class labels as positive and negative.

Before moving to derive the model, we need little basics of linear algebra.

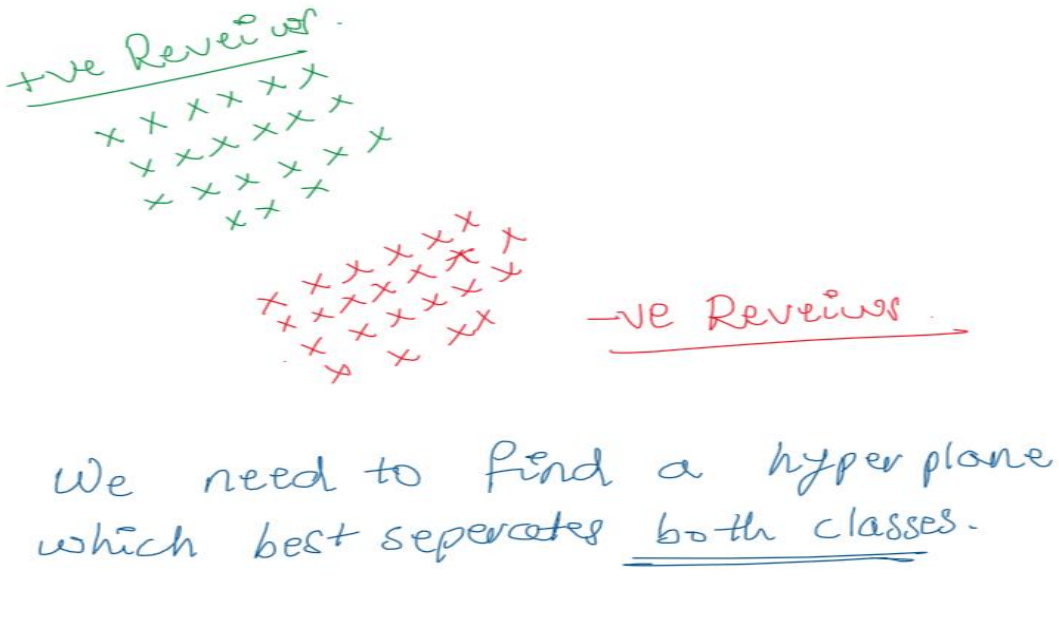


Distance of a point from a hyperplane

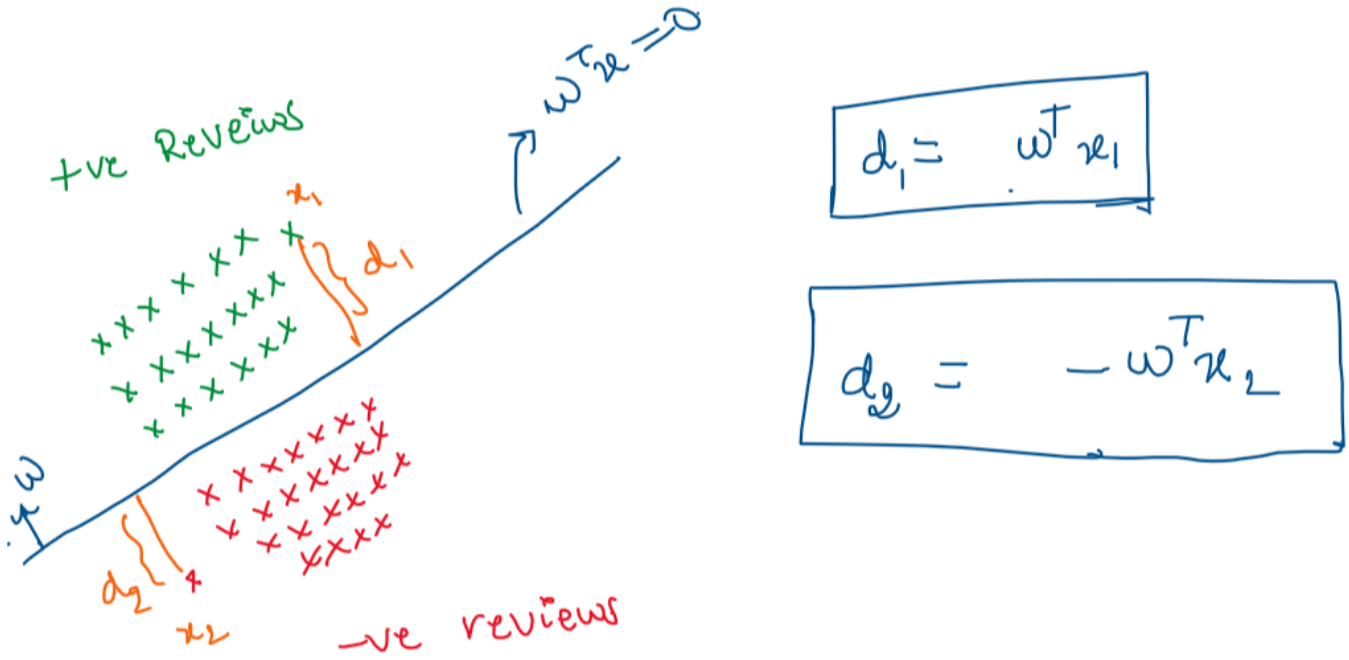
w is the normal vector to the hyperplane.

p is *d*-dimensional point.

Now consider the vector representation of all the reviews.



So, here the assumption is, data points are linearly / almost linearly seperable.



We need to find a hyperplane/ decision boundary like this.

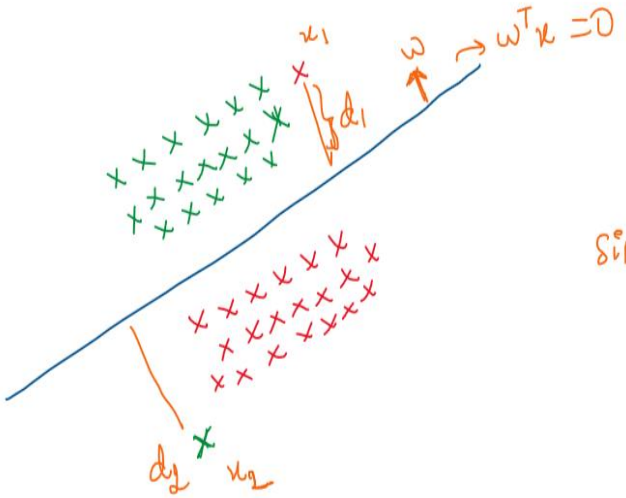
How to find the hyperplane/Decision boundary?

For that all we need is value for w vector. If we find the vector w, we can draw hyperplane.

To find the vector w, we should formulate optimization problem.

In our case, target variable is either +1 or -1.

Hence it is clear from the below diagram that, $y_i(w^T x_i) > 0$ For correctly classified points. Similarly, $y_i(w^T x_i) < 0$ For incorrectly classified points.



$$d_1 = w^T x_1 > 0$$

Since $y_1 = -1$,

$$y_1 (w^T x_1) = -w^T x_1 < 0$$

$$d_2 = -w^T x_2 < 0$$

Since $y_2 = +1$,

$$y_2 (-w^T x_2) = (-w^T x_2) = w^T x_2 < 0$$

For any other points which are correctly classified,

$$y_i (w^T x_i) > 0$$

So, our goal is to maximize correctly classified points and minimize incorrectly classified points.

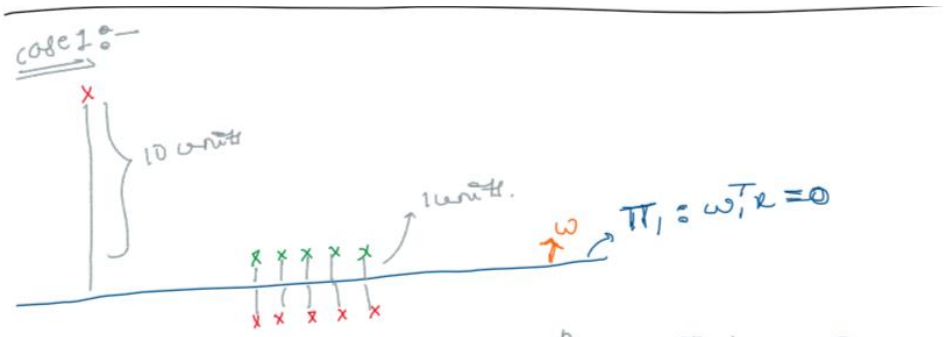
The optimization problem becomes,

optimization problem :

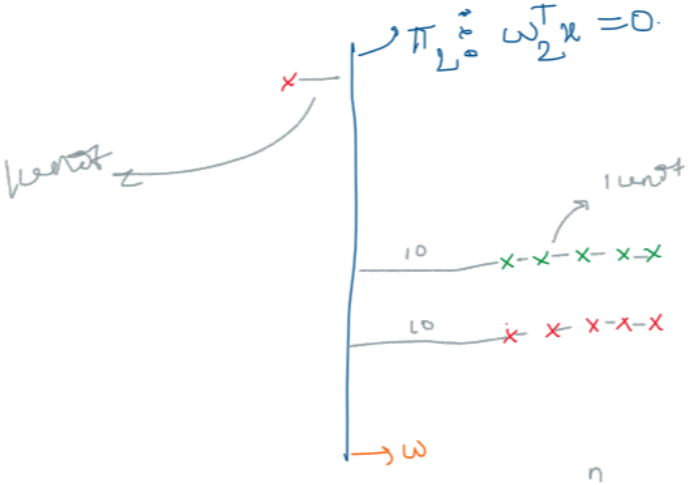
$$w^* = \underset{w}{\operatorname{argmax}} \sum_{i=1}^{n} y_i (w^T x_i)$$

This is maximum when all points are correctly classified.

But there is a problem with this equation. This equation is very much prone to outliers. Let us see that,



Now, if we calculate, $\sum_{i=1}^{n} y_i (w^T x_i) = 0$



Now, $\sum_{i=1}^{n} y_i (w^T x_i)$

$= (10+11+12+13+14) - (10+11+12+13+14) + 1$

$\sum_{i=1}^{n} y_i (w^T x_i) = 1$

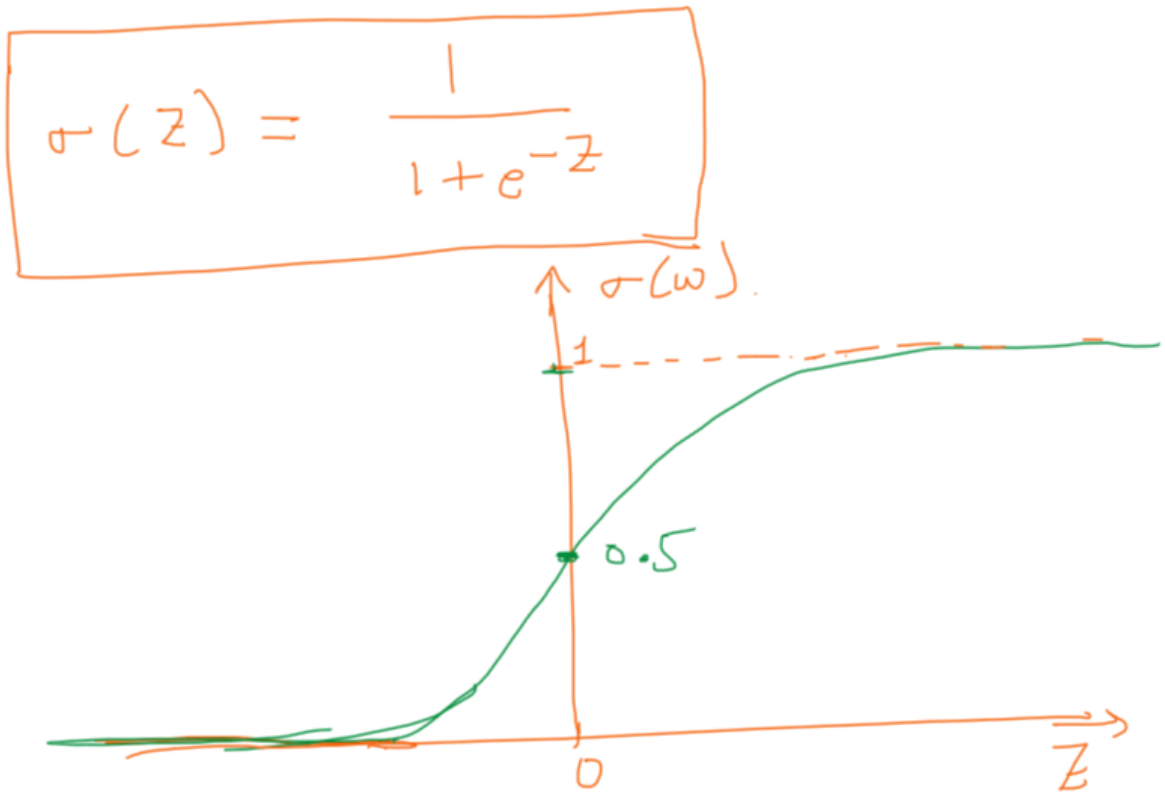
According to our optimization problem, we should choose 2nd hyperplane over first hyperplane. But we can clearly see that 2nd hyperplane is worst compared to 1st hyperplane. Because misclassifications in 1st hyperplane is just 1. In case of 2nd hyperplane, it is 5.

This is due to one outlier point.

So, we have to modify the optimization problem, such that large distances should be squashed and smaller distances should be considered as it is.

Also, function should be easily differentiable.

One such function is sigmoid function.



We can use sigmoid function in our optimization problem. Now the optimization problem becomes,

$$w^* = \arg\max_w \sum_{i=1}^{n} \frac{1}{1+e^{-y_i(w^T x_i)}}$$

$$\Rightarrow w^* = \arg\max_w \sum_{i=1}^{n} \log\left[\frac{1}{1+e^{-y_i(w^T x_i)}}\right]$$

$$\Rightarrow w^* = \arg\max_w \sum_{i=1}^{n} \log(1) - \log\left(1+e^{-y_i(w^T x_i)}\right)$$

$$\Rightarrow w^* = \arg\max_w \sum_{i=1}^{n} -\log\left(1+e^{-y_i(w^T x_i)}\right)$$

$$\Rightarrow \boxed{w^* = \arg\min_w \sum_{i=1}^{n} \log\left(1+e^{-y_i(w^T x_i)}\right)}$$

This is the optimization problem.

So, by solving this we get best w vector. Using that w, we can find hyperplane which classifies the given points.

This term is called as log loss.

We can also derive this using probability and distribution.

By deriving from probability, we will get another form of log loss.

$$\text{log loss} = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(p_i) + (1-y_i)\log(1-p_i)$$

where $y_i = +1$ or 0

$$p_i = \frac{1}{1+e^{-w^T x_i}}$$

In case of geometric derivation, y = +1 or -1

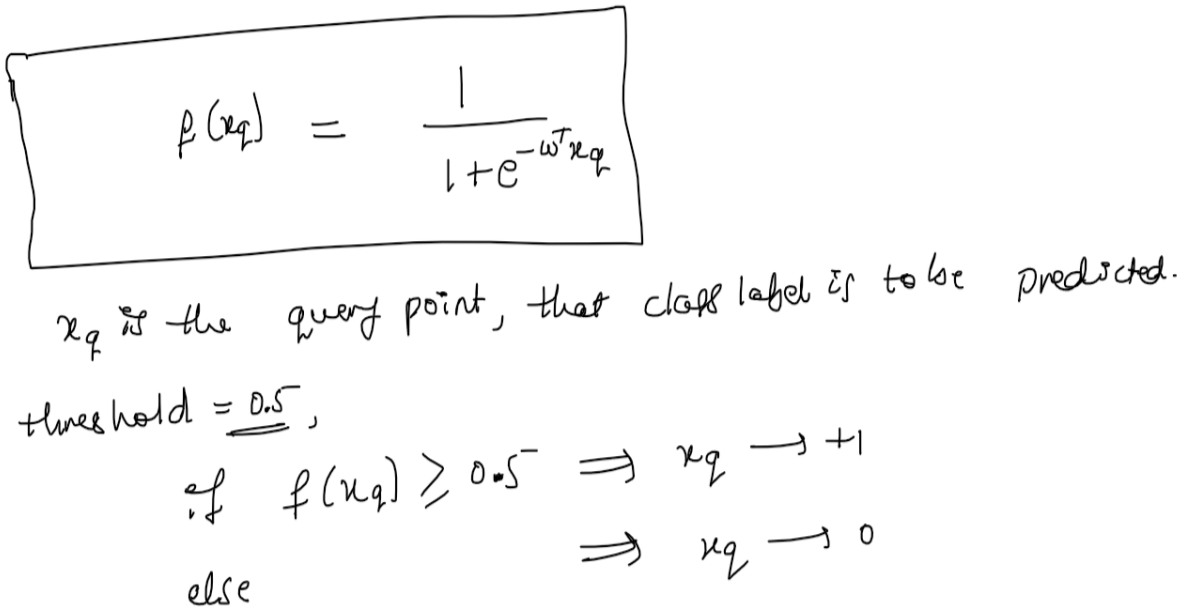
In case of probabilistic derivation y = 1 or 0.

Using gradient descent, we can solve this objective function.

SGD is better than simple GD if we have large data.

Also, we can add regularization to avoid overfitting.

So, after finding the best hyperplane, i.e., weight vector w, we can easily find the class label for given query point. For that we just need to pass that query point to sigmoid function. By passing to the sigmoid function, we get the probability like score. By using some threshold (like 0.5) we can predict the class label as 0 or 1.



Advantages of Logistic regression:

- It is simple to implement.
- It will perform very well if we have large data also.
- It is easy to interpret the results. That is interpretability is good.

Limitations:

- If we have non linearly separable data, LR will fail.

Logistic regression performs very well in real world if we have almost linearly separable data.

Also, there are many more classifiers which are used in real world for classification. LR is one of the simplest models.

References:

- Vijay K. Rohatgi and A. K. Md. Ehsanes Saleh (2015) An Introduction to Probability and Statistics. John Wiley & Sons, Inc.
- https://en.wikipedia.org/wiki/Logistic_regression
- Sebastian Raschka and Wahid Mirjalili (2017) Python for Machine Learning
- Probabilistic approach: https://medium.com/nerd-for-tech/logistic-regression-a-probabilistic-approach-633da06f216

THANK YOU